

REMARKS**Drawings**

Formal drawings are herein submitted under separate cover addressed to Official Draftsperson. Also included is a set of the drawings marked in red to indicate changes. The changes are to add or change element per Examiner's request. Examiner is requested to approve changes noted in red.

Specification

Element numbers are added in the specification in the amendments set out herein above so that the specification corresponds to the element numbers in the drawings.

Element numbers are deleted from the abstract in the specification amendments set out herein above.

Informalities pointed out in the First Office action are corrected in the specification amendments set out herein above.

Support for Claim Amendments

Claim amendments submitted herein are for a variety of reasons, as will be discussed herein below. Support for the amendments will first be described.

Support for the amendments to claim 2 and 10 is as follows. The collection of live maps is formed on a client. FIG. 2; specification, page 5, lines 18 through 19; specification, page 7, lines 6 and 7. A live map includes identification of a transaction for actual processing and data required for the application. Specification, page 5, lines 18 through 29; specification, page 6, lines 14 through 16. The chosen computing application (of the transaction identified in a live map) is the same for all the live maps. Specification, page 7, lines 9 through 11. Measuring performance results from the server actually processing the load, and the measuring may be performed by either the client or the server. Specification, page 6, lines 14 through 16; specification, page 7, lines 14 through 19. Changing the live maps includes changing the types of transactions identified therein. Specification, page 6, lines 24 through 25.

Support for the amendment to claims 6 and 10 regarding use of the terms "workstation," "client," and "client emulation server" is as follows. The "workstation" is also referred to in the present application as a "client." Specification, page 4, lines 2 and 3; specification, page 5, lines

32 through 34. A number of clients may be emulated by a client emulation server. Specification, page 5, lines 32 through 34.

Support for the amendment to claim 6 regarding measuring alternatively by the server is as follows. The measuring may be performed by either the client or the server. Specification, page 6, lines 14 through 16; specification, page 7, lines 14 through 19.

Claims 4, 5, 15 and 16 are amended to refer to the term "load" in a manner consistent with the amendment to claims 2 and 10, since the amended claims 2 and 10 refer to a first processing load and a next processing load.

Claim 12 is amended to improve clarity and antecedent basis.

Claim 13 is amended to eliminate use of the term "datastore," which term, as Examiner has pointed out, is not used in the specification.

Claim 14 is amended to eliminate use of the term "output," as requested. Also, claim 14 is amended to set out the alternative in which the client stores the performance data measures instead of the server. Support is included in the specification for this change to claim 14 at page 7, lines 14 through 19.

Claims 17 and 18 are amended to eliminate use of the term "application server." Although this term is suggested in the specification, since the server runs a chosen application; nevertheless, the term is not explicitly used in the specification. Claims 17 and 18 are also amended to state that the database servers are "operable" to execute portions of the load transactions, since this language is more consistent with claims in the apparatus form.

Claim 18 is amended to replace "formed by" with -- comprises -- for consistency and clarity.

Claim Objections

Claims 9 and 23, which were objected to due to informalities, are herein canceled.

Claims 19 and 22 were objected to due to the use of the word "datastore." Claim 19 is canceled and claim 22 is amended herein above to eliminate the word usage.

Claim Rejections 35 USC 112

Claim 14, which was rejected due to the use of the word "output," is amended herein above to eliminate the word usage.

Claim 21, which was rejected due to multiplicity, is herein canceled.

To overcome the rejection for indefiniteness in claims 1 through 27, claims 1 and 10 are herein amended to incorporate explicit language about "live maps."

Claim Rejections 35 USC 102

Claims 1 through 9 have been rejected as being anticipated by U.S. patent 6,446,028 ("Wang"). The amendments set out herein above are submitted to overcome these rejections.

According to amended method claim 1 (and system claim 10, as well) of the present application, a client forms a collection of live maps which identify transactions and include data required for the transactions, and sends the collection to a server. The same client measures performance of the server. Alternatively, the server measures its own performance.

Wang discloses a third computing system that monitors performance of a server, or performance of a client-server interaction. See Wang FIG. 6, for example. Furthermore, Wang specifically discusses shortcomings of, and thus teaches away from, prior art in which clients have stub code so that a client can measure performance of a server with which the client interacts. Wang, column 4, lines 1 through 67 ("Prior Art Performance Monitoring").

Also, while Wang discloses that the client sends packets to the server, these are the well known TCP/IP packets. Wang does not suggest that the client forms a specific *collection* of information (referred to in the lexicography of the present application as "*live maps*") identifying transactions and including data required for the transactions. On the contrary, Wang teaches that it is advantageous that "... the only change made to the network is the addition of the main performance measurement monitor 490." Wang, column 5, lines 18 through 20. This is contrasted to "... code placed at each client system and each server system ..." Wang, column 5, lines 21 through 22.

Applicant requests that the above distinctive and advantageous features of the amended claims be given due weight. While Wang discloses that code may be placed at a client system and a server system for monitoring, Wang does not teach how, and actually teaches away from doing this. The collection of live maps as per amended claim 1 (and 10) in the present case, is advantageous because it enables performance measurement of actual transactions. As shown,

results are advantageously mapped to transactions. Specification, pages 9 through 11. See also FIG. 5.

Note also, Wang does not suggest that the transactions of the collection are all for the *same computing application*, as now in amended claim 1 (and 10) of the present application. The performance measurement is further facilitated by this feature of the present invention, since all the transactions for a measured processing load are for the same application running on the server. It is difficult to account for uncontrolled load variations. This homogeneity tends to reduce uncontrolled variations.

In another aspect of claim 1 (and 10) of the present application, as amended, the first collection of live maps (i.e., the collection on the client) is changed and transmitted from the client to the server, so that the server processing load resulting from the first collection of live maps is different than the load resulting from the next, changed collection. More specifically, the changing includes changing the number of live maps and types of transactions in the first collection of live maps. Then, the transactions for the new live maps are actually processed by the server and performance measuring is repeated.

These features of the amended claims are also significant, advantageous and deserving of weight. Specifically, these features address the problems described in the "Background of the Invention" in the present case regarding actual execution instead of feeding simulated transactions to a server. Furthermore, the iterative process and the limiting of the collection of maps to a single application advantageously enables exploration of server performance as performance responds to selective variation in the nature of the processing load. Specification, pages 9 through 11. Wang does not suggest this.

Claim Rejections 35 USC 103

Claims 10 through 27 have been rejected as being obviated by U.S. patent 6,446,028 ("Wang") in view of U.S. patent 5,812,780 ("Chen"). The amendments set out herein above are submitted to overcome these rejections.

Chen was cited in the prior Office Action with respect to claim 10 of the present application for Chen's teaching concerning a single computing system emulating multiple systems. The amendment set out above incorporates features into claim 10 which now

patentably distinguish the present invention, as discussed immediately above herein. That is, as per amended claim 10, a client forms a collection of live maps which identify transactions and include data required for the transactions, and sends the collection to a server. The same client measures performance of the server. Alternatively, the server measures its own performance. Also, the transactions of the collection are all for the same computing application. Further, the first collection of live maps (i.e., the collection on the client) is changed and transmitted from the client to the server, so that the server processing load resulting from the first collection of live maps is different than the load resulting from the next, changed collection. More specifically, the changing includes changing the number of live maps and types of transactions in the first collection of live maps. Then, the transactions for the new live maps are actually processed by the server and performance measuring is repeated. Neither Chen, nor Chen in combination with Wang teach or suggest these features.

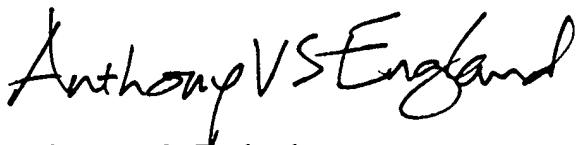
PRIOR ART OF RECORD

Applicants have reviewed the prior art of record cited by but not relied upon by Examiner, and assert that the invention is patentably distinct.

REQUESTED ACTION

Applicants have amended claims for allowance in accordance with Examiner's objections and hereby request that Examiner grant allowance and prompt passage of the application to issuance. Attorney hereby requests Examiner to call Attorney to discuss the case.

Respectfully submitted,



Anthony V. S. England
Attorney for Applicants
Registration No. 35,129
512-477-7165



In the Specification

Paragraph beginning at page 5, line 10:

RECEIVED
JAN 31 2003
Technology Center 2100

Fig. 2 is a generalised software architecture for a client-server environment. On the client machine, a Graphical User Interface (GUI) layer 210 provides the human-machine interface for a user. The GUI layer 210 interfaces with an application layer 220, where the specific computing operation or purpose performed by the client-server system resides. The application layer 220 interfaces with a middleware layer 230 that handles system aspects such as system resource usage, operating system locks, shared memory access, container services, queueing Services, transaction Services, logical unit of work coordination, inter-process communications, user access control services and configuration retrieval services. As shown, application data, packaged into " maps " or " containers " 250, is passed to the middleware layer 230. The middleware layer 230 represents the operating system and communications services. The transport layer 240 of the client machine is in network communication with the server machine. The server machine replicates the layers 240, 230 and 220, providing a replica transport layer 280, the replica middleware layer 270, and replica the application layer 260, and functions thereof.

Paragraph beginning at page 5, line 24:

The content of a map/container 250 includes the identification of the " service " which the server machine application is to execute, together with the application data which is required by the particular application process. Fig. 3 shows a representative data packet 310 having header information 320 specific to the transport and middleware layers 240 and 230 (Fig. 2). Optionally, there can be similar trailer information 340. The maps/container content 330 comprises the services information and application data.

Paragraph beginning at page 6, line 1:

Fig. 4A shows an example of a server machine 100, emulating a client machine, in networked connection with a server machine 102 that is to be stress-tested.

Paragraph beginning at page 6, line 14:

(i) The ~~of~~ live maps/containers for a plurality of transactions for a chosen application must firstly be collected. By "live" is meant actual transactions, as opposed to simulations.

Paragraph beginning at page 7, line 1:

In the pre-runtime 120, a Business Workload Definition File 501 is created and populated, creating 502 a Business Workload Distribution File 503. This file 503 and a Mmapping Ffile 505 (mapping Business Transactions To Machine Transactions 505) are merged to create 504 the machine workload, resulting in a Machine Workload Execution Definition File 506. In the run-time 122, the pre-stored Llive Mmaps 510 are selectively read by a Mmap Ssending Pprogram 511 which executes the Workload Execution File 506 to place the process load onto the server 102 running the application under test. The Map Sending Program 511 is replicated: one per client machine being simulated. The server 102 under test executes the requested load and returns a reply map. Such reply maps are stored on the emulated client machine in the Maps Received File 512. It is necessary for the Business Workload Definition File 501 and the Mapping File 503 to relate to the same application that is being run by the server 102 under test. In the same way, the stored maps in the Maps Received File 512 must relate to the same server application.

Paragraph beginning at page 7, line 14:

The performance criteria, such as the average response time of a transaction or the proportion of CPU time taken by a transaction, can be determined by the server under test 102

itself, or can be determined on the client emulation server (to include the communications link performance). Whichever way, the results of the performance testing are stored in a Logging File 515.1 on the client emulation server or on the server under test Logging File 515.2.

Paragraph beginning at page 7, line 21:

An example of the Business Workload Definition File 501, for a Telco customer enquiry and ordering system (such as as generally described above) is as follows:

Paragraph beginning at page 8, line 1:

An example of the file 505 which maps Business Transactions (of sub-type DA) to a sequence of maps to be executed is as follows:

Paragraph beginning at page 8, line 16:

An example of Machine Workload Execution ~~D~~definition ~~E~~file 506 is as follows:

Paragraph beginning at page 9, line 9:

Referring again to Fig. 2, as examples of implementations for the middleware layers 230 include the IBM CICSTM or ENCNIATM systems. In relation to the transport layer 240, examples of implementations are either TCP/IP or SNA. Any convenient physical layer network can be utilized, such as a token passing LAN. The application layer 220 must have the capability, either inherently or by specific coding, to create or write live maps.

Abstract:

A method for testing server machine performance is described. A client-emulating server machine(100) has a collection of live data maps for a plurality of transactions for a chosen computing application. A server(102) is in communication with the workstation(100). The workstation(100) transmits a processing load, including ~~constituted by~~ a plurality of the maps for the plurality of transactions, to the server(102) as it executes the computing load. The server(102) measures one or more performance criteria as it executes the load. The performance criteria can include the average response time for a transaction within a load, and the proportion of server CPU time taken by each transaction of the load. By varying the processing load generated by the workstation(100) and assessing the measured performance criteria, it is possible to determine whether the server (102) has satisfactory capacity.

In the Claims

1. (cancelled)

2. (once amended) A method for testing performance of a server running a chosen computing application, comprising the steps of:

(a) forming on a client a first collection of a number of live maps, wherein such a live map includes i) identification of a transaction for actual processing of the transactions by the server running a chosen computing application, and ii) data for the chosen application, and wherein the chosen computing application of the transaction for such a live map is the same for each of the live maps in the collection;

(b) transmitting a first processing load from the client to the server running said computing application, wherein the processing load includes the first collection of the number of said live maps for a plurality of said transactions;

(c) measuring one or more performance criteria resulting from said server server actually processing said load, wherein the measuring is performed by the client or the server; and

The method of claim 1, comprising the further step of:-

(d) changing the first collection of live maps and transmitting a next processing load from the client to the server, the next processing load including the changed collection of live maps, varying in order to selectively vary said processing loads, wherein the changing includes by making changes to the number of said live maps and the mixtypes of said transactions in the first collection of live maps transmitted to said server,; and whereby wherein said measuring step (c) is repeated for each the next individual processing load.

4. (once amended) The method of claim 3, whereby said performance criteria include average response time for a transaction within such a load.

5. (once amended) The method of claim 3, whereby said performance criteria include the proportion of server CPU time taken by a each transaction of such a ~~said~~ load.

6. (once amended) The method of claim 1, wherein step ~~A~~ ~~method for testing server performance, comprising the steps of:~~

——— ~~(a) forming a collection of live maps for a plurality of transactions for a chosen computing application;~~

——— ~~(b) transmitting a processing load, constituted by a plurality of said maps for a plurality of transactions, from a workstation to a server running said computing application;~~

——— (c) comprises, for each transaction within said load, returning a result to said client ~~workstation~~; and ~~(d) measuring, by at said client workstation, the~~ one or more performance criteria responsive to the processing ~~based on execution~~ of said load by said server.

7 through 9. (cancelled)

10. (once amended) A system for testing server performance, said system comprising:

(a) a server running a chosen computing application;

~~_____~~ (b) a client workstation sized to representing a plurality of individual client computing stations, said workstation client including a data store of a first collection of a number of live maps, wherein such a live map includes i) identification of a transaction for actual processing of the transactions by the server running the chosen computing application, and ii) data for the chosen application, and wherein the chosen computing application of the transaction for such a live map is the same for each of the live maps in the collection for a plurality of transactions for a chosen application;

~~_____~~ (b) ~~a server running said chosen application; and~~

(c) a communications connection between said workstation client and said server; and
~~_____~~ wherein said workstation client is operable to transmit a first processing load to said server, via said communications connection, the processing load including the first collection constituted by a plurality of said live maps for a plurality of said transactions, and said server is operable to actually process said load, said server or client, but not necessarily both the server and client, is operable to measures one or more performance criteria resulting from as the server it executes processing said load, and wherein said client is further operable to change the first collection of live maps and transmit a next processing load to the server, the next processing load including the changed collection of live maps, in order to selectively vary said processing loads, wherein the changing includes changing the number of said live maps and types of said transactions in the first collection of live maps, and the first server or client is operable to repeat the measuring for the next processing load.

11. (cancelled)

12. (once amended) The system of claim 10~~11~~, wherein said server compares said measured performance criteria against predetermined performance measures to determine whether the server has satisfactory~~its capacity is satisfactory~~.

13. (once amended) The system of claim 12, wherein said server ~~maintains a data~~ stores a file of said performance data measures.

14. (once amended) The system of claim 13, wherein said client~~server~~ stores a file of ~~produces an output representing~~ said performance data measures.

15. (once amended) The system of claim 12, wherein said performance data criteria includes the average response time for a transaction within one of said loads.

16. (once amended) The system of claim 12, wherein said performance data criteria includes the proportion of server CPU time taken by such a ~~each~~ transaction of said loads.

17. (once amended) The system of claim 12, wherein said ~~application~~ server has connection to one or more database servers, said database servers being operable to execute~~ing~~ portions of said load transactions.

18. (once amended) The system of claim 12, wherein said ~~application-server~~ comprises ~~is~~ ~~formed by~~ a plurality of servers, and each of said server plurality has connection to one or more database servers, said database servers being operable to executing portions of said load transactions.

19 through 21. (cancelled)

22. (once amended) The system of claim 10, ~~A system for testing server performance,~~
said system comprising:

- ~~— (a) a workstation sized to represent a plurality of individual client computing stations,~~
~~said workstation including a datastore of a collection of live maps for a plurality of transactions~~
~~for a chosen application;~~
- ~~— (b) a server running said chosen application;~~
- ~~(c) at least one database in communication with said server; and~~

23 through 27. (cancelled)